A1307-300-SRS-2

28 August 1997

SOFTWARE REQUIREMENTS SPECIFICATION
FOR THE
CONTROL, DISPLAY, AND REDUCTION
COMPUTER SOFTWARE CONFIGURATION ITEM
THREAT SIMULATOR LINKING ACTIVITIES NETWORK

CONTRACT NO. F08635-92-C-0050

CDRL SEQUENCE NO. (A053)

Prepared for:
Joint Advanced Distributed Simulation Program Office
Joint Test Force
Kirtland AFB, NM 87117

Prepared by:
Georgia Institute of Technology
Georgia Tech Research Institute
Atlanta, Georgia 30332

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

1.1. IDENTIFICATION

Linking of existing test facilities is seen as a method for enhancing the realism and thoroughness of Test and Evaluation (T&E) for Electronic Warfare (EW) assets. These individual test facilities, for the most part, already exist. The linking is to be accomplished through the communication services available from the commercial long-haul carriers which also already exist. The network interactions are to be carried out according to the interface specifications defined by the Defense Modeling and Simulation Office in the High Level Architecture Interface Specification. This Software Requirement Specification identifies the requirements for the Control, Display, and Reduction Software Segment (CDRSS) of the Threat Simulation Linking Activities (TSLA) Network which will support the implementation of Advanced Distributed Simulation (ADS) as required to perform T&E of EW assets in realistic combat environments.

1.2. COMPUTER SOFTWARE CONFIGURATION ITEM OVERVIEW

The CDRSS Computer Software Configuration Item (CSCI) is responsible for providing the control, display, data reduction, and data analysis functions of the TSLA network. Figure 1 describes the TSLA network architecture. The shaded regions of this network architecture diagram indicate the functions addressed by this CSCI. The CDRSS shall exist in two specific configurations. The configuration located at the Test Director Facility (TDF) shall support all network-level functions including monitoring of all network entities data and health status and controlling state transitions of each entity on the real-time network. The Node Executive (NE) configuration shall support all entity level functions including monitoring local entity data and health status, controlling entity states, and reporting data and status information to the real-time network. Except where explicitly noted, all definitions within this specification shall apply to both the TDF and NE configuration of the CDRSS.

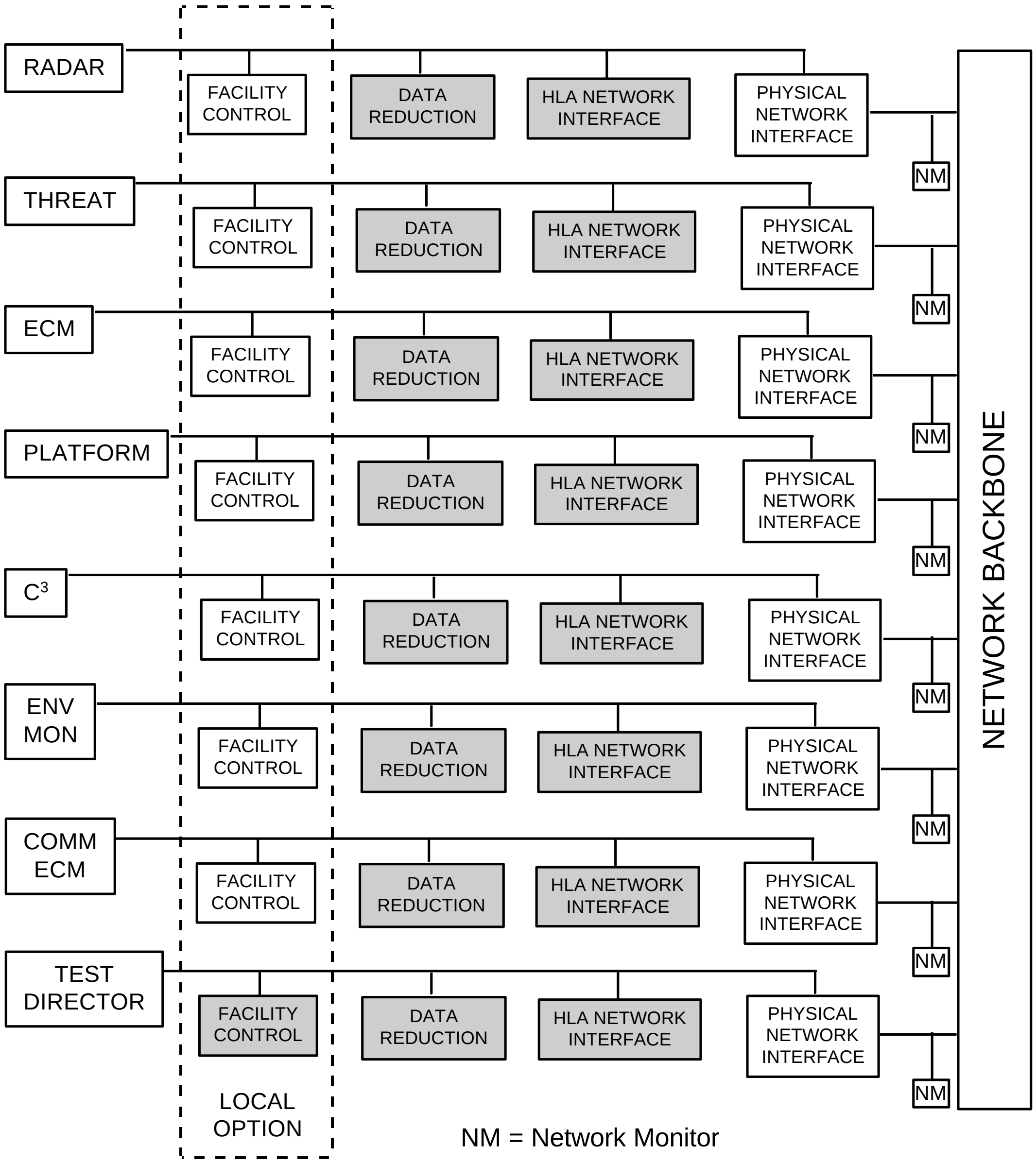

Figure 1. Threat Simulation Linking Activities Network Interface Architecture Diagram

1.3. Document Overview

This document defines the necessary software requirements to provide the capabilities necessary to implement the control, display, and reduction function of the TSLA network.

2. APPLICABLE DOCUMENTS

2.1. GOVERNMENT DOCUMENTS

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

2.1.1. SPECIFICATIONS

1. MIL-T-31000 - General specifications for Technical Data Package Lists
2. High Level Architecture Rules, version 1.0, dated 15 Aug 1996
3. High Level Architecture Interface Specification, version 1.1, dated 12 Feb 1997
4. High Level Architecture Object Model Template, version 1.1, dated 12 Feb 1997
5. Network Requirements Specification for Threat Simulator Linking Activities in support of EW Testing, 22 April 1997.
6. A-1307-300-SSS-1 - System Specification for the TSLA Network 27 June 1997, or current version
7. A-1307-300-IRS-1 - Interface Specification for the TSLA Network 27 June 1997, or current version

2.1.2. STANDARDS

1. MIL-STD 1521B, Notice 1 - Technical Reviews and Audits for Systems, Equipments, and Computer Software
2. MIL-STD-129K, Notice 1 - Marking for Shipment and Storage
3. MIL-STD 481B - Configuration control - engineering Changes (short form), Deviations and Waivers
4. MIL-STD-483A - Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs
5. MIL-STD-882B, Notice 1 - System Safety Program Requirements
6. MIL-STD-100E - Engineering Drawing Practices
7. World Geographic Society Standard (1984) MIL-STD-2401 World Geodetic System, WGS-84.
8. MIL-STD-1472D, Human Engineering Design Criteria For Military Systems, Equipment and Facilities
9. MIL-H-46855B, Human engineering Requirements for Military Systems, Equipment and Facilities

2.1.3. OTHER

1. AF-MAN- 99-112: Air Force Electronic Warfare T&E Process Manual, 27 March 1995.
2. Feasibility Study Report, “Joint Advanced Distributed Simulation,” Prepared by ODDR&E (T&E), Final Report, February 1995.
3. DoD 5220.22M - Industrial Security Manual for Safeguarding Classified Information

2.1.4. DRAWINGS

None

2.1.5. OTHER PUBLICATIONS

None

2.2. NON-GOVERNMENT DOCUMENTS

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

2.2.1. SPECIFICATIONS

None

2.2.2. STANDARDS

1. Information technology--Local and metropolitan area networks--Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, ANSI/IEEE Std 802.3, 1996 Edition
2. Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 2: Logical link control, ANSI/IEEE Std 802.2, 1994 Edition.
3. Synchronization Interface Standards for Digital Networks (ANSI T1.101-1994)
4. Digital Hierarchy - Electrical Interfaces (ANSI T1.102-1993)
5. Exchange-Interexchange Carrier Interfaces-Individual Channel Signaling Protocols (ANSI T1.104-1991)
6. Digital Hierarchy - Formats Specifications (ANSI T1.107-1995)
7. IEEE-STD-1278.1 - Standards for distributed Interactive Simulation (DIS) -- Application Protocols, 1993.
8. Current InterNIC standards on TCP/IP protocol and document in current Request For Comments (RFC) Documents. At the time of this specification, the current RFC was 1780.

2.2.3. DRAWINGS

None

2.2.4. OTHER PUBLICATIONS

None

3. ENGINEERING REQUIREMENTS

3.1. CSCI EXTERNAL INTERFACE REQUIREMENTS

There are three external interfaces associated with the CDRSS CSCI. Figure 2 illustrates both the internal interfaces between the five capabilities of the CDRSS and the external interfaces of the CDRSS

CSCI. Section 3.2 will define the CDRSS capabilities and Section 3.3 will describe the internal interfaces of the CDRSS. This section shall describe the external interfaces with respect to the CDRSS.

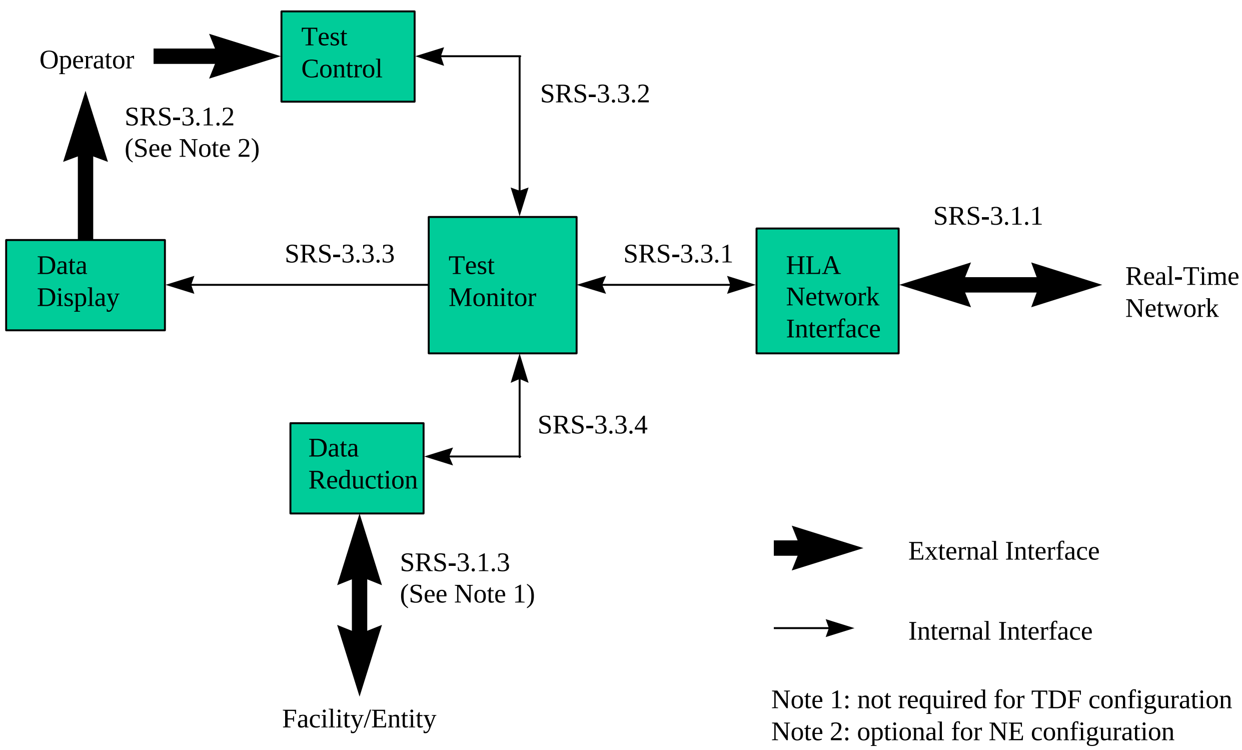

Figure 2. Control, Display, Reduction Software Segment Interface Diagram

3.1.1. REAL-TIME NETWORK INTERFACE

This interface is responsible for all data communication between the entity and the real-time network. All data associated with the encapsulation of the entity (refer to Section 3.4 and the Appendices for data encapsulation descriptions) shall be transmitted over this interface. The protocol for data transmission shall conform to the High Level Architecture (HLA) interface specification. The specific data interaction descriptions are contained in the TSLA Interface Requirement Specification.

3.1.2. OPERATOR INTERFACE

This interface supports the human operator interaction with the real-time network entity. This interface shall be implemented as a graphical user interface (GUI) designed for real-time applications. Emphasis shall be placed on displaying information in a format that can quickly and easily be detected by the operator, and a simple menu hierarchy to permit quick access to control commands. The display and control requirements for the CDRSS CSCI are described in Section 3.2 and 3.11.

3.1.3. FACILITY/ENTITY INTERFACE

This interface is responsible for receiving the raw data from the facility or entity instrumentation that is necessary to perform the local data reduction function and to provide data required by other entities on the real-time network. The requirements for types of data and transmission protocols for this interface shall be driven by the individual facility or entity interface specifications and control documents. The interactions with the CDRSS shall be limited to the data requirements given in Section 3.4 and the Appendices. This interface will not be required for the TDF.

3.2. CSCI CAPABILITY REQUIREMENTS

The CDRSS CSCI consists of five capability requirements as shown in Figure 2, (1) Test Monitoring, (2) Test Control, (3) Data Reduction, (4) HLA Network Interface, (5) and Data Display. Each of these capabilities are described in the following sections. The Test Control Capability (TCC) should not be considered synonymous with the TDF. The TCC represents a capability requirement of the CDRSS which is a software segment executing at the TDF as well as the NEs. The TDF is a physical entity on the real-time network. References to the operator in the following sections of this specification refer to the human interaction with the CDRSS. This operator includes both the TDF operators as well as the NE operators unless specifically stated otherwise in the following sections.

3.2.1. TEST MONITOR CAPABILITY (SSS 3.1.1.9)

The CDRSS CSCI is responsible for monitoring all test activities. The CDRSS monitors data to (1) assess the health status of the ADS network, (2) analyze the quality of test data, (3) analyze the quality of the test environment and test assets, (4) evaluate the performance and/or effectiveness of the System Under Test (SUT), and (5) perform data logging of data reduction processing inputs and outputs.

3.2.1.1. INITIALIZATION MODE (SSS 3.1.1.9, 3.2.1.1, 3.2.1.2)

During network initialization the CDRSS monitors information about the health of the network and entities connected to the network. The NE configuration located at the local entities is responsible for collecting entity self-test and ready-for-test information, which on demand by the TDF configuration can be transmitted during the network initialization process. After voice confirmation from all network entities that they are ready to join, The TDF configuration is responsible for sending commands to the HLA Run-Time Infrastructure (RTI) to create the federation and initiate publication of TDF attributes. The TDF shall then command each network entity to join the federation and establish its publication/subscription requirements with the RTI. The TDF shall then be responsible for collecting the Roll Call responses from each entity and providing the necessary information to the TDF operator relative to network status. The network status information at a minimum shall contain results from facility or entity Built In Test (BIT) procedures, numbers of checksum errors encountered over the network during federation creation and initialization, the revision level of the Federation Required Execution Details (FRED), RTI, and Application Program Interface (API), and results of latency measurements. Initialization Mode shall also provide a Standby Condition which will permit the TDF operator to place all entities in Standby Condition, ready to begin a test trial. Once all network entities have entered the Standby

Condition, the TDF shall initiate a test scenario to verify publish/subscription requirements across the network. This test scenario shall verify that all subscribed attributes are being published and the network entity-to-network entity data transfer latencies are within network performance tolerances.

3.2.1.2. EXECUTION MODE (SSS 3.1.1.9, 3.2.1.1, 3.2.1.3, 3.7.9.2.1).

The CDRSS Test Monitoring Capability is utilized during execution mode to capture all data collected and processed during the test. The CDRSS must be capable of monitoring the network status information described in Section 3.2.1.1, the quality of test data, and the SUT performance data. Data monitored by the CDRSS during test execution shall be logged to permit replay of any test at sufficient fidelity to analyze test results relative to each of the test objectives. The precise content and location of the data logging shall be entity dependent based on the test objectives.

Maintenance functions shall be performed within the federation execution. The purpose of maintenance activities is to determine that network entities are HLA compliant and that requirements for network reliability, latency, and connectivity are being met. These functions shall be performed while operating in the execution mode. The specific maintenance procedures shall be assessed in preparation of lower level development specifications.

3.2.1.2.1. QUALITY-OF-TEST DATA (SSS 3.1.1.9)

Quality-of-test data provides operators the information to assess how the test is proceeding. This assessment requires accurate visualization of the scenario, quality assurance (QA) information for the SUT, and network health status. This information shall be monitored and made available for presentation to the operators in real-time. This will allow for quick corrective actions in the instances where problems are identified with either the network, a test entity, or the test design.

3.2.1.2.1.1. SCENARIO VISUALIZATION DATA

Scenario visualization data shall include information to support a pictorial representation of the scenario map and player interactions. This shall include a geographical description of the test area, Time-Space-Position Information (TSPI) for all non-stationary test assets, and operating modes versus times for all appropriate test entities (i.e., threat simulators, SUT). The specific requirements for the geographical description are test dependent. Typically an EW test will require a description of the terrain, location of landmarks, location of fixed test assets, identification of restricted regions within the test space, threat engagement /lethal envelopes, and if appropriate a latitude/longitude grid for location of the simulated test area. This information shall be distributed to the participants prior to the test. TSPI data shall be monitored for each non-stationary test asset at the full data rate available from the network. This is necessary to assure that all movements of test entities are captured at sufficient fidelity to support post-test analysis. All threat mode state information shall be recorded to allow characterization of the RF environment presented to the SUT during the test. Since response time related test objectives are common in EW testing, the full resolution and accuracy of threat mode state change times shall be recorded as well. The time the state change was initiated as well as the time the state change was received at a subscribed entity shall be recorded for post-test analysis. By a similar argument, SUT mode state changes and their associated time must also be recorded at the available resolution and accuracy.

3.2.1.2.1.2. QUALITY ASSURANCE DATA

QA data includes information that confirms the actual test environment is the same as the expected test environment or identifies when differences exist. The scenario visualization data provides QA relative to player positions. Of equal importance, is confirmation that the known players are operating in expected modes and within specific parameter limits and that any unintentional player interactions have been identified and characterized. QA data includes a comparison of the commanded SUT responses with the measured responses. For a jammer, the commanded SUT response would be the ECM (Electronic Countermeasures) technique descriptor or ID. The measured SUT response would be the ECM waveforms received by instrumentation monitoring the RF emissions from the jammer. QA data for the command and control functions shall consist of data typically associated with a filter center plot board. The filter center plot board provides a graphical overview of the command center in a command and control network. The data necessary to generate this plot board, target tracks and target assignments, shall be monitored to determine how well the command and control entities are performing. The Test Monitoring Capability shall monitor data which verifies that known emitters are radiating within expected test parameters and which indicates intervals of time when they are outside specified limits. In addition, data relevant to characterization of unknown environment emissions shall be considered a part of the QA data monitoring function. All QA data related to the test environment shall be monitored and recorded by the TDF configuration of the CDRSS at the resolution, accuracy, and rate available over the network.

3.2.1.2.1.3. NETWORK QUALITY DATA

Another important piece of the quality-of-test data includes the quality of the data transmitted over the real-time network. Errors in data transmission, if not identified quickly, can result in useless test data being analyzed later for final results. The CDRSS must be capable of monitoring the health of the network relative to results of Built-in-test procedures for network objects, as well as for network links, error rates in data transmissions through validation of checksum or other error detection mechanisms, and latency characterization of data transmitted over the network. The NE configuration of the CDRSS shall be responsible for network quality data monitoring relative to the entity only. The TDF configuration of the CDRSS shall be responsible for network quality data monitoring for all network entities. This shall be accomplished by each network entity on the network reporting results of self-tests and network quality tests in the form of a health status message. This message at a minimum, shall indicate that the entity is capable of responding to commands over the network as well as communicate information back to the TDF CDRSS. Additional requirements for detailed status of the objects health shall be entity dependent. All network packets shall be encoded with error-detection coding. The recommended coding scheme should utilize a checksum word for error-detection, however equivalent codes are allowable provided they are capable of detecting single and multiple bit errors in the data. There is no requirement for an error-correction coding scheme for the real-time network. All network transmissions shall be time-tagged at the source where the attribute is first created, not when it is transmitted. Likewise, all latency computations shall use the time that the data was consumed by the simulation, not when it was received at the facility. This is necessary to determine the total latency of the data from one entity to another. As each packet is received, the data reduction capability described in Section 3.2.3 shall compute a message latency based on the received time and the time tag in the data packet. This information shall be monitored by the CDRSS.

3.2.1.2.2. SYSTEM UNDER TEST DATA (SSS 3.1.1.9)

SUT data to be monitored by the CDRSS shall include all information necessary to evaluate the performance of the test item. Since the precise requirements for this data are test-item dependent, this SRS identifies the general requirements for different classes of SUTs. SUT data is typically obtained from specialized instrumentation that is connected to the test article. For the real-time network, this instrumentation will reformat the SUT data, perform data reduction and publish it over the network to the other objects that require the data. The NE configuration of the CDRSS shall be responsible for monitoring this local data collection and reduction process, and publishing the data. The TDF configuration of the CDRSS shall be responsible for monitoring this NE-generated data for each SUT entity reporting on the real-time network and computing the final Measures of Performance (MOP) and/or Measures of Effectiveness (MOE). The classes of EW test assets include RF receivers, jammers and towed decoys, expendables, and integrated EW mission processors. The types of data necessary from each of these systems are summarized below.

Additional data may be required depending on the test requirements. The data listed below identifies minimum requirements for each SUT class.

RF receivers - Receiver mode, receiver display files/display status, threat track files, sensor counts vs. time (missile warning receivers)

Jammers, towed decoys - All RF receiver data requirements (if jammer is equipped with a receiver processor function), commanded technique vs. threat, jammer mode/status

Expendables - remaining expendable count/amount, deployment time, expendable type, deployment method (pilot, radar warning receiver/jammer commanded, etc.), emitter track file(when available), number of expendables deployed

Integrated EW mission processor - All data necessary for each EW asset connected to the mission processor, detection reports vs. time, commanded responses vs. time, mission processor mode.

3.2.1.3. POST-TEST MODE (SSS 3.1.1.9, 3.2.1.1, 3.2.1.4)

During post test, the TDF shall be responsible for monitoring all status information reported by each entity on the real-time network. This information shall include BIT results, checksum errors, and latency measurements described in the previous section as well as the post-test diagnostic information or required test details. The TDF configuration of the CDRSS shall be responsible for monitoring these reports as each entity is polled during the post-test roll-call process. Detailed post-test data that is required at a location other than the location where it is generated shall be transmitted electronically by normal methods outside of the federation such as FTP or disk. The specific contents and formats of data logs for post-test analysis shall be defined as part of the federation development process.

3.2.2. TEST CONTROL CAPABILITY (SSS 3.1.1.9)

The TCC is similar for the TDF and NE configurations, however, some differences exist as to the provider and user of control information. The TDF is responsible for controlling all network activities during a test. This shall include (1) configuring the network and initializing each network entity, (2) controlling the execution of the test via start/resume, stop, pause, rewind, and playback commands, and (3) performing post-test status queries for each entity utilized during the test. The NE is responsible for controlling all entity activities and responding to TDF commands during a test. This shall include (1) performing all necessary state transitions under TDF command, (2) providing status/health and test data to the TDF, and (3) responding to test execution commands from the TDF.

3.2.2.1. INITIALIZATION MODE (SSS 3.1.1.9, 3.2.1, 3.2.1.1, 3.2.1.2)

The TDF TCC shall be responsible for initializing the network nodes and establishing a network status prior to test execution. Immediately after federation initialization, the test director shall determine the status of each entity through a roll-call procedure. As each entity is polled, a ready or not ready status shall be reported back by the NE TCC to the TDF TCC. The TDF TCC shall maintain a state identifier for each entity. This state identifier shall indicate (1) if the entity is waiting to be polled, (2) the entity has been polled, but needs to be re-polled due to a not ready status being reported, or (3) the entity has been initialized and ready for start of test execution (stand-by mode). Additional states shall be tracked by this identifier as described in the following sections under test execution and post-test modes. Once all entities have entered stand-by mode as indicated by the state identifiers maintained by the TDF, a Start Test Command may then be issued to begin test execution.

3.2.2.2. EXECUTION MODE (SSS 3.1.1.9, 3.2.1, 3.2.1.1. 3.2.1.3)

The CSCI shall be capable of controlling the progress of the test from start to finish. The TDF operator shall be able to execute commands for starting the test and stopping the test. Depending on the test requirements, the TDF operator shall also have the capability to pause, rewind, or playback during test execution. Each of these capabilities are described below. Maintenance functions shall be controlled by the TCC capability as well. The specific requirements for TCC relative to maintenance functions shall be assessed in preparation of the lower level development specifications.

3.2.2.2.1. START TEST

A Start Test Command shall be executed by the TDF operator at the beginning of a test, or to restart a test that has been paused. Prior to the start of a new test, the TDF TCC shall wait until all test assets have entered the stand-by mode. At this time, the TDF TCC shall notify the TDF operator that the network is ready for a Start Test Command. Once initiated, all test assets will be given a command to begin along with the time that the Start Test Command was issued. This will be necessary to synchronize each network asset to a common start-of-test time. If a test has been paused, the TDF operator shall be notified that the network is in a paused state and is awaiting a test restart command. Once the command is given, the CSCI shall transmit the same message to all test assets as the Start Test Command, however, the time reported will be the time that the restart command was issued.

3.2.2.2.2. STOP TEST

A Stop Test Command is executed by the TDF operator once at the end of test. A Stop Test Command issued by the TDF operator shall result in the TDF TCC issuing a Stop Test Command to each network entity. This will result in the network entering the post-test mode and begin the status reporting process associated with this state. A Stop Test Command may be issued by the TDF operator whenever a test is in the execution mode. The TDF TCC shall also have the capability to initiate an automatic Stop Test Command at a pre-defined test termination time or event.

3.2.2.2.3. PAUSE TEST

A Pause Test Command may be executed by the TDF operator whenever a test is actively running in execution mode. A Pause Test Command is required prior to execution of a Rewind or Playback command. When a Pause Test Command is issued by the TDF operator, the TDF TCC will send a command out over the test network to suspend all test execution. All constructive and virtual entities will reset to their state at the time the pause command was issued. This time shall be included within the pause command issued by the TDF TCC. Live entities shall also reset to their state at the time the pause command was issued and if possible remain in this state until another command is issued. For those live entities such as aircraft that cannot be paused in this manner, the closest realizable state shall be maintained (e.g., an aircraft should fly a tight circle around the point in space when the pause was issued). The state of all entities shall be updated and reported to the TDF CDRSS for monitoring purposes.

3.2.2.2.4. REWIND

A Rewind Command may be issued by the TDF operator only after a pause command has been given. The TDF operator shall issue a rewind command to return to a previous point in the test scenario. The TDF TCC shall report the updated scenario time to each test asset. Each asset shall be responsible for returning to the correct state at the desired time issued in the command. The TDF operator shall have the option to remove or include accumulated analysis results collected prior to the rewind command. If the TDF operator selects to remove this data, each entity shall be responsible for removing all accumulated samples between the rewind time and the current test scenario time. If the TDF operator selects not to remove the data, the data collected during the rewind interval will be accumulated as new data to the existing test data. A Test Start Command must be issued by the TDF operator to restart test execution from the rewound position.

3.2.2.2.5. PLAYBACK

A Playback Command may be issued by the TDF operator only after a Pause Command has been given. A Playback Command will allow the TDF operator to replay any portion of the test currently executing. While in playback, only the previously recorded data will be observed. No new data will be generated by any network entity. The TDF TCC shall accept either a time or test event from the TDF operator to begin the playback process. This event or time must be prior to the current test execution time. Playback will begin at the specified time or event and continue until the current test execution time is reached or the TDF operator terminates the playback operations. The TDF TCC shall issue a Playback Command over the network to each of the network entities. Copies of data local to the TDF CDRSS shall be used to support playback to the maximum extent possible. Data not available at the TDF CDRSS shall be requested by the CSCI on an as-needed basis and at the time it is required. Variable playback speeds shall be supported from real-time speeds down to a manual single-step mode under control of the TDF TCC. The playback speed shall be defined by the TDF operator. When possible, faster than real-time playback would also be desirable.

3.2.2.3. POST-TEST MODE (SSS 3.2.1, 3.2.1.1, 3.2.1.4)

The TCC after issuing the Stop Test Command and entering the post-test mode shall command each network entity to enter post-test mode. As each network entity is commanded to enter post-test mode the NE TCC shall return pertinent information related to the test execution. This shall include general entity health status information (i.e., number of entity failures during the test if any, network transmission errors, and amount of data available for downloading for post test analysis), and a report of test anomalies to include out-of-bound conditions, deviations from the original test design, and any data necessary for post-test analysis. After transmitting this information, each NE CDRSS may then resign from the network at any time. The TDF TCC shall maintain the state of all network entities and report any that fail to respond to the post-test query to the TDF operator.

3.2.3. DATA REDUCTION CAPABILITY (SSS 3.2.3.4)

The CDRSS is responsible for performing all real-time and post-test data reduction. Due to the volume of data that potentially may be generated in an ADS environment, distributed data reduction at each network facility is a requirement. This distributed data reduction provides a real-time capability that would not be possible with a centralized data reduction function. The primary functions of the data reduction capability shall be to (1) assess the status and health of the ADS network and (2) evaluate the performance of the SUT. Partitioning of the SUT performance evaluation between the TDF and NE configurations shall permit the generation of the data elements provided in Section 3.4 and the Appendices. The NE Data Reduction Capability shall process local entity data to reduced technical performance parameters (TPP) and system performance parameters (SPP). These TPPs and SPPs shall be used by the TDF Data Reduction Capability to derive the SUT MOPs and MOEs. Section 3.2.3.2.2 defines the TPPs and SPPs associated with each SUT MOP and MOE.

3.2.3.1. INITIALIZATION MODE (SSS 3.2.1.1, 3.2.1.2, 3.2.3.4)

During initialization, the TDF Data Reduction Capability shall be responsible for processing the entity reports received by the TDF during the Roll-Call and Pub-Sub Connectivity Conditions. During the roll-call function, the network entity reports shall be analyzed to identify the readiness of each entity and that current versions of the FRED, RTI, and API are being used. The results of this analysis shall be reported to the TDF operator via the Data Display Capability described below. During the Pub-Sub Connectivity Condition, the CDRSS Data Reduction Capability shall analyze network entity reports to determine the condition of the network with respect to connectivity, error rate performance, and latency.

3.2.3.2. EXECUTION MODE (SSS 3.2.1.1, 3.2.1.3, 3.2.3.4, 3.7.9.2.3.2)

As discussed below, two levels of Data Reduction Capabilities are needed during test execution. These are the NE Data Reduction Capabilities that can be performed at each network entity, and the TDF Data Reduction Capabilities that determine overall network health status and SUT performance. Each of these capabilities is discussed separately. All data reduction capabilities are needed to support maintenance functions as well.

3.2.3.2.1. NODE EXECUTIVE DATA REDUCTION CAPABILITY (SSS 3.2.3.4)

At a minimum, each entity or Facility Control for an entity (ies) shall provide local test data reduction capabilities for any data that is generated or consumed by that entity. The raw data shall be stored locally as a backup or to serve as an off-line diagnostic tool. Data reduction shall be required to generate the necessary SPPs and TPPs as follows:

1. Reference sensor entities shall report the measured position of entities in earth-centered coordinates according to the WGS84 standard. Any coordinate transformation from the native sensor coordinate system to the earth-centered system shall be performed by the Data Reduction Capability.
2. In some instances, it is necessary to compensate for the latency in transmission of data. In these cases, a filter-predictor shall be used to operate on the raw time-tagged data. The predictor shall provide extrapolations of the current position data based on the state variable estimates produced by the filter. Specifications for the filter-predictor are provided in Appendix D.
3. All tracking sensors shall compute their own tracking error by comparing the target location reported by that sensor with that reported by the reference sensor entity.
4. In those instances where the weapon entity is implemented in a simulation, miss distance shall be computed by comparing the trajectory of the simulated weapon to the position of the target entity as reported by the reference sensor. The vector point of closest approach in 3-dimensional space shall be reported as the miss distance. The miss distance shall be reported in meters.
5. All sensor data shall be reported in the units and with the resolution specified in Appendices B and C. Any conversion of raw sensor data needed to convert to these standard units shall be performed by the NE Data Reduction Capability.
6. The NE Data Reduction Capability shall provide local diagnostic capabilities to determine the health of the local entity and the local connection to the network.

3.2.3.2.2. TEST DIRECTOR FACILITY DATA REDUCTION CAPABILITY (SSS 3.7.9.2.3.3)

The TDF Data Reduction Capability shall compute SUT performance measures as required for the test. These SUT performance measures fall into 3 primary categories for EW testing. They are: RF receiver measures, jammer measures, and expendables measures. The general requirements for each of these categories are the same. The TDF shall process monitored test data to derive each MOP/MOE applicable to the SUT in real-time when possible, post-test otherwise. Post-test processing shall be considered only when the real-time data is considered inadequate for accurate performance computation (i.e., the real-time data products require post-test QA processing before being used in performance calculations), or the real-time data volume transfer rate exceeds the bandwidth of the real-time network. The MOP/MOE derivation by the TDF Data Reduction Capability shall be limited to the processing of the SPPs and TPPs defined in Section 3.2.3.2.1. The computation of the SPPs and TPPs shall be performed by the NE Data Reduction Capability as discussed in Section 3.2.3.2.1. The SUT MOP/MOEs and their definitions are as follows:

RF Receiver MOPs -

1. Response Time for Signal Intercept - determine the time necessary for the RF receiver to detect a signal after an emission is activated.
2. Threat Detection/Deletion Range - determine the range at which the signal strength drops below the RF receiver sensitivity and results in the RF emission not being detected.
3. Correct Emitter Identification - determine that an RF receiver correctly characterizes and identifies (ambiguously or non-ambiguously) an RF emission.
4. Response Time for Correct Final Identification - determine the time necessary for the RF receiver to characterize or identify (ambiguously or non-ambiguously) after an emission is activated.
5. Correct Threat Prioritization - determine the ability of the RF receiver to correctly prioritize a multiple set of detected RF emissions in accordance with SUT system specifications.
6. Direction Finding Accuracy - determine the ability of the RF receiver to correctly measure the angle-of-arrival of an RF emission.

Jammer MOPs -

1. ECM Technique Response Time - determine the time necessary for a jammer to generate a response after an emission is activated and the time after an emission is detected by the receiver processor of the ECM system if available.
2. Correct Technique Selection - determine the percentage of time the correct jammer response is selected by the jammer for a particular RF threat.
3. ECM Technique Generation, ECM Technique Activation/Deactivation - determine the time and range when a jammer response is generated with respect to the engagement envelope for the threat.

4. Multi-Threat Jamming Response - determine the ability of a jammer to generate multiple responses against multiple RF threats. Determine that correct resource allocation is occurring for a power managed and/or resource limited jammer system.
5. Correct Technique Parameter Emission - Determine that the technique selected by the jammer system is radiating correctly with the correct parameters and the correct synchronization in both frequency and time with the RF threat.

Jammer MOEs -

1. Jamming-to-Signal Ratio - Evaluate/compare the Jamming-to-signal ratio of ECM responses under user-defined test events or conditions.
2. Engagement Time/ Envelope Reduction - evaluate the ability of the jammer to reduce the amount of time and the distance available to an RF threat to engage and track the test aircraft reliably.
3. Tracking Error - Evaluate/compare the tracking error of an RF threat in the presence of jamming and without jamming.
4. Break-lock/ Track Loss - Evaluate the increase in occurrences of terminal threat track loss in the presence of jamming compared to no jamming .
5. Reduction in Missile Launches/Projectiles Fired - evaluate the ability of the jammer to reduce the number of opportunities a threat system is able to fire a missile or a gun at the test aircraft.
6. Vector Missile Miss Distance - Compute the point of closest approach to the target for each missile launch. The miss distance shall be reported as a vector, not a single magnitude value to more accurately assess the impact of the weapon.
7. Reduction in Lethality - compute the reduction in lethality created by the use of jamming based on the number of missile/projectile hits/misses with and without ECM.
8. Probability of Kill - Compute the probability of a kill from a missile launch or gun fire against the test aircraft.
9. Net Reduction in Lethality - compute the reduction in lethality created by the use of jamming based on probability of kill with and without jamming.

Expendables MOPs -

1. All Jammer MOPs
2. Dispense Rate - Evaluate the minimum time between deployment of multiple sets of expendables.
3. Dispense Response Time - Determine the time necessary to dispense an expendable from threat activation, threat detection, and time dispense command is issued.
4. Correct Dispense Program - Determine the percentage of time the correct dispense program was executed against each threat.

Expendables MOEs -

1. MOEs 2-9 from Jammer MOPs

In addition to SUT MOP/MOEs, the TDF Data Reduction Capability shall also compute the following network performance measures:

1. Network Bandwidth - compute maximum, minimum, and average bandwidth requirements. Process real-time bandwidth requirements for displaying bandwidth vs. time during the test
2. Network Error Rates - compute the number of network bit errors as well as dropped message packets vs. time.
3. Network Latency - compute the minimum, maximum, and average message latency over the network. Process real-time latency measurements for displaying vs. time during the test.

3.2.3.3. Post-Test Mode (SSS 3.2.1.1, 3.2.1.4, 3.2.3.4)

At the completion of a test, each entity shall be required to complete any data reduction functions not completed in real-time. If post-test data reduction results are required at a location other than the location where they were generated, the data shall be transmitted by normal means (e.g., FTP, disk, etc.). Defining the specific contents and formats of this post-test data is part of the federation development process. Local copies of all reduced and raw data shall be archived for future analysis functions. The TDF Data Reduction Capability shall accept all post-test data from each entity and archive the data as required for future analysis. In addition, all MOP/MOEs described in Section 3.2.3.2 shall be updated with any post-test data in accordance with the test plan.

3.2.4. HLA Network Interface Capability (SSS 3.2.3.2)

The HLA network interface capability shall conform to the HLA Interface Specification cited in Section 2. Since the TSLA network includes live entities, real-time test execution is required. There are no requirements for the time management services as specified in the HLA Interface Specification.

The HLA interface shall be responsible for publishing and subscribing to the current values for each data element specified in the Appendices for the given entity class to be interfaced. If the HLA Interface is being developed for a facility serviced by a Facility Control, this HLA Interface shall provide this function for all entities managed by the Facility Control.

3.2.5. Data Display Capability (SSS 3.7.9.2.4)

The CDRSS CSCI shall provide real-time controls and displays for the test. This capability shall be implemented with a reconfigurable, GUI to allow the operator to quickly interpret displayed data, make decisions, and execute commands. This GUI shall support resizable windows that contain color images and text that can enable the operator to assimilate more information and perform with higher productivity. The Data Display Capability shall support reconfigurable displays including specifying the number of windows, the size of each window, the location of the window, and the type of data or menu displayed. The Data Display Capability shall also support the capability to select axis scaling for all displays. In particular, axis scaling includes changing the scenario map region (e.g., zoom in, zoom out, move display), changing the time resolution on displays, and changing the axis resolution on the performance measure displays.

3.2.5.1. INITIALIZATION MODE (SSS 3.2.1.1, 3.2.1.2, 3.2.3.4, 3.7.9.2.3.1)

The Data Display Capability shall provide a status window that will display the state of each entity on the TSLA network. The TDF configuration shall display the state of all network entity states, while the NE configuration shall only display the individual entity state. This display shall provide sufficient information to all operators to determine the readiness of each test asset. At a minimum this status display shall contain the time of day, the state of each network entity, and any error messages that result from the initialization of the network. Error messages shall be contained within a scrolled region within the status window, to allow the operator to review multiple error messages. The error message shall include the entity identifier which reported the error, the severity of the error (i.e., fatal error versus warning condition) and sufficient textual information to permit the operator to make a go/no go decision for the test.

3.2.5.2. EXECUTION MODE (SSS 3.2.1.1, 3.2.1.3, 3.2.3.4, 3.7.9.2.3, 3.7.9.2.3.1, 3.7.9.2.3.2)

During test execution, the Data Display Capability shall be required to support several different displays to provide adequate information to the operator. These displays fall into two categories: (1) displays associated with test visualization and awareness, and (2) displays associated with SUT performance measures. The test visualization and awareness displays include:

1. Status Display
2. Scenario Map
3. SUT Display
4. Filter Plot Board Display
5. Emitter Activity Display
6. Technique Activity Display
7. SUT Performance Measure Display

The requirements for these displays as well as the SUT performance measures displays are described in the following sections.

3.2.5.2.1. STATUS DISPLAY

The status display provided during the execution mode shall be identical to the status display for the initialization mode with the addition of the test start time (wall-clock time) and the elapsed test time.

3.2.5.2.2. SCENARIO MAP

A map of the region (either real or simulated) where the test is executing shall be provided at the TDF. At a minimum this map shall display the geographical region including terrain, physical landmarks, latitude/longitude grid, threat positions and state (off, search/acquisition, track, missile launch), SUT position and attitude vs. time, and the position and attitude of all mobile entities such as Airborne Interceptors. Color will be used to indicate different states of entities. Different symbols will also be used for different types of entities to allow the test director the ability to quickly differentiate the players within a test scenario. A sample of this display window is shown in Figure 3.

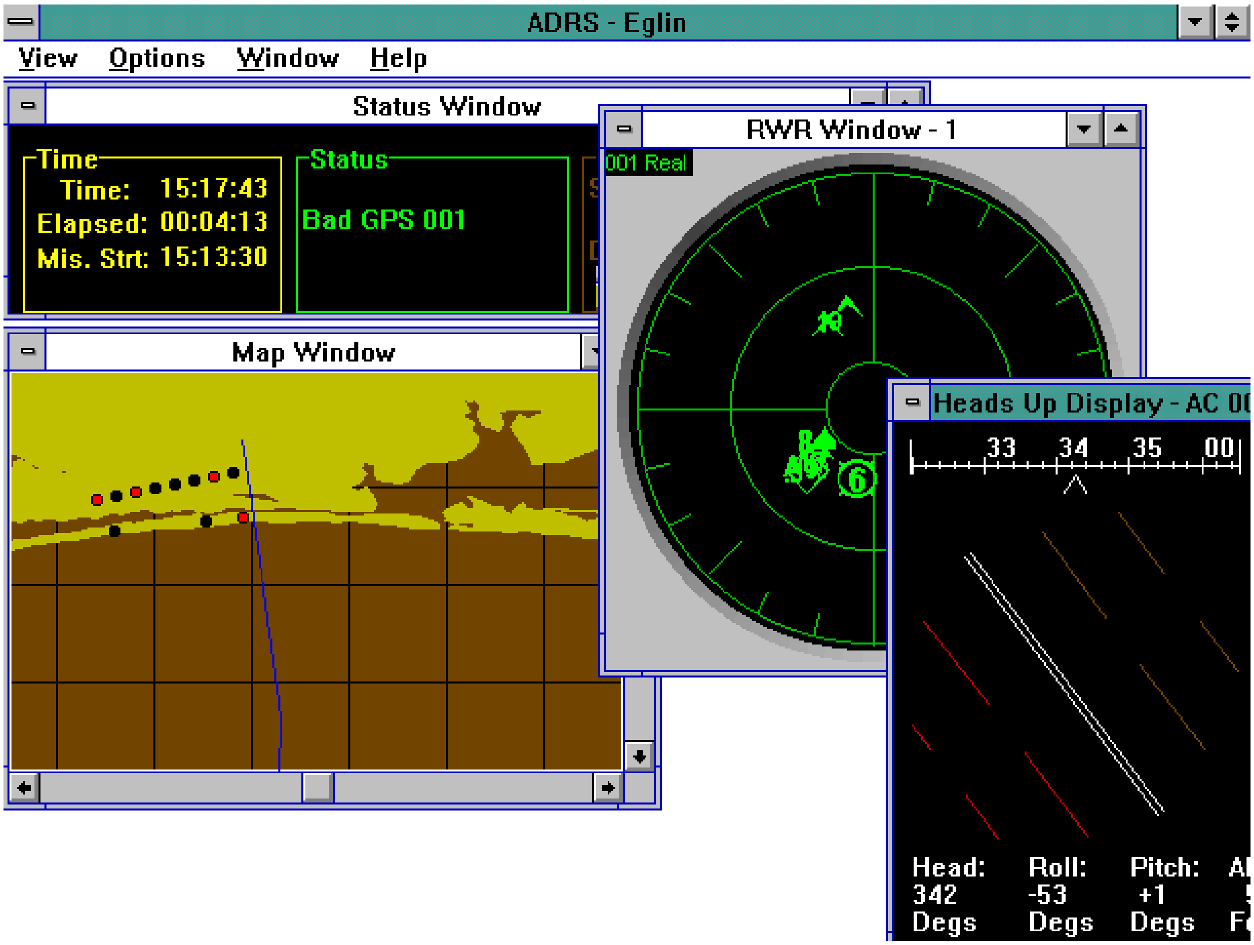

Figure 3. Example Scenario Map Display

3.2.5.2.3. SUT Display

A display window shall be available at the TDF which provides a reasonable representation of the SUT display as seen by the SUT operator. This SUT display shall have the capability to display the actual information reported by the SUT during the test, as well as the expected information based on pre-test predictions if available. The TDF operator shall have the capability to toggle the SUT display between actual only, expected only, or both. In the event, multiple SUTs of a similar type are being evaluated, the CDRSS shall support overlay of information on the same SUT display. This overlay capability shall be limited to a maximum of two sets of SUT information. If more than two SUTs of a particular type exist during a test execution, the TDF operator shall have the capability to select the two sets of data to be overlaid. In addition to the display that replicates the SUT operator's display, an additional display shall be provided that shows SUT responses versus time in the form of a color bar chart. An example of this display is given in Figure 4. Different colors shall be used to indicate different states of SUT responses. When applicable, states of entities in the scenario that caused the SUT response shall also be displayed on this bar chart.

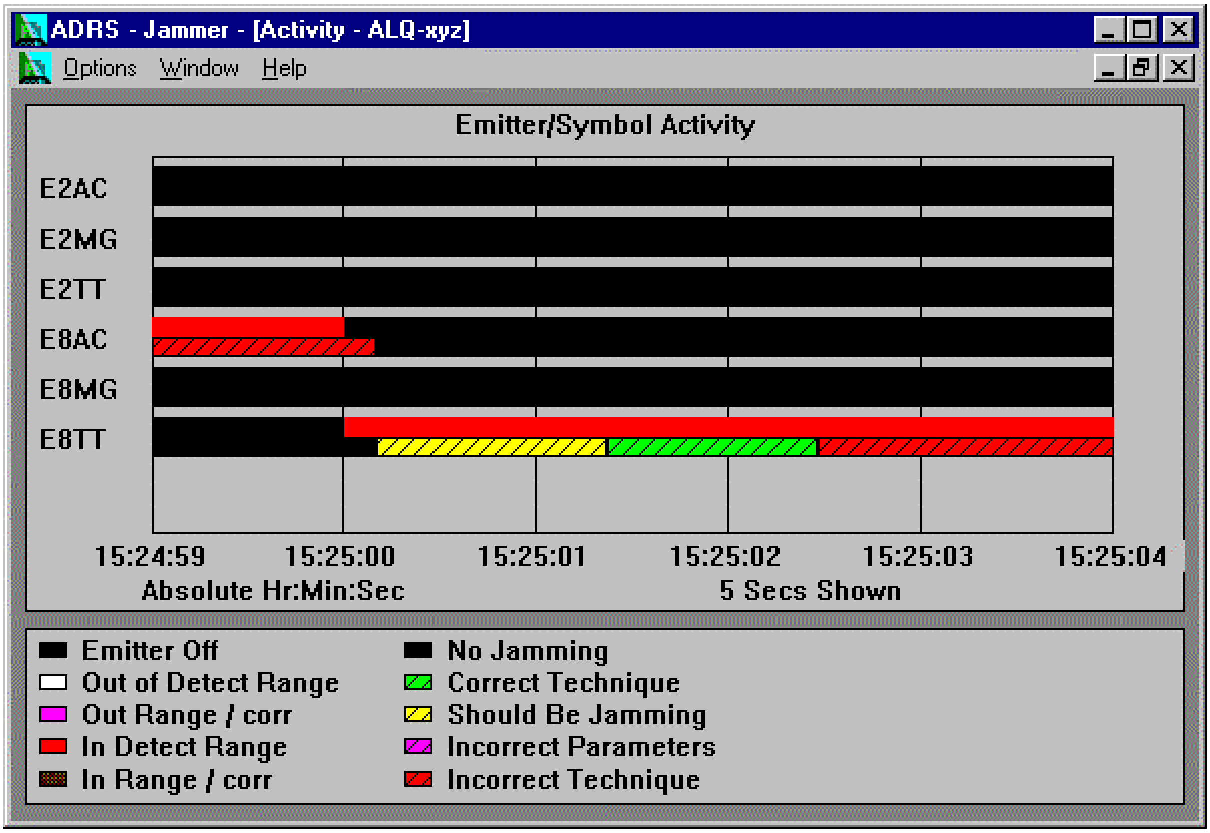

Figure 4. Example System Under Test Activity Display

3.2.5.2.4. FILTER PLOT BOARD DISPLAY

The TDF Data Display Capability shall display the status of C^3 assets in the form of a filter center plot board as shown in Figure 5. This display shall provide a 2-dimensional plot of target tracks versus time and the assignment status of each track. This assignment status shall indicate the track ID, the terminal threat assignment ID (or none if no threat assigned), and the kill/ no kill status of the target.

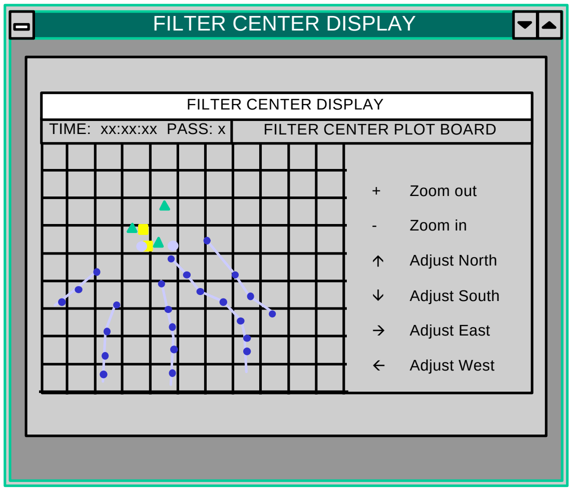

Figure 5. Example Filter Center Plot Board Display

3.2.5.2.5. Emitter Activity Display

A tabular list of all active emitters shall be provided by the TDF Data Display Capability. This display shall provide a list of all emitters currently radiating in the scenario. Each emitter shall be displayed in two modes - commanded state and verified state. As each emitter is activated or changes states via script or test control command, its expected state shall be displayed in the emitter activity display using the same color designations as those used on the scenario map. QA instrumentation reports shall provide verified states of each of these active emitters. The verified state shall also be displayed in this tabular emitter activity display. A non-verified state shall be indicated on this display as well.

3.2.5.2.6. Technique Activity Display

A tabular list of actively radiating ECM techniques shall be provided by the TDF Data Display Capability. This display shall provide a commanded and verified state for each active ECM technique similar to the requirements for the Emitter Activity Display. In addition, the technique activity shall be linked to the emitter that the ECM is intended to defeat. This shall be done by providing an additional column in the Emitter Activity Display indicating the assigned technique. Color designations shall be used to indicate commanded or verified status of the ECM technique listed for each emitter.

3.2.5.2.7. System-Under-Test Performance Measure Display

SUT performance data shall be provided in graphical displays. The TDF Data Display Capability shall support at a minimum x-y graphs, polar plots, histogram (single and multiple bars), and contour plots. Other graphs may be required based on SUT MOP/MOEs and display requirements identified by the test objectives. The SUT performance displays shall support the selection of the type of graph, the MOP/MOE, and the data condition (i.e., emitter/threat/technique grouping, test condition, test time interval) by the TDF operator. The display shall be updated real-time as data is made available from the real-time network. Rescaling of the graphs shall be supported for all coordinate axis during real-time test execution. The MOP/MOEs supported by this TDF display were listed in Section 3.2.3.2.2. The ability of TDF to display these measures does not imply the requirement to compute all components of these measures at the TDF as described under the Data Reduction Capability of the CDRSS.

3.2.5.2.8. DATA REDUCTION DISPLAYS

As required by an entity, intermediate data reduction results shall be displayed by the NE Data Display Capability. These intermediate results are entity dependent. The requirements for these displays shall be derived from entity specifications and documentation.

3.2.5.3. POST-TEST MODE (SSS 3.2.1.1, 3.2.1.3, 3.2.3.4, 3.7.9.2.3.1)

The Data Display Capability shall provide a status window that will display the state of each entity on the TSLA network during the post-test activities. As during initialization modes, the TDF configuration shall display the status of all network entities, while the NE configuration shall only display the status of the individual entity. This display shall provide sufficient information to the operators to determine the status of each test asset during the post-test roll-call function. This status shall include any information reported by the network entity relative to test anomalies and test event execution. At a minimum this status display shall contain the time of day, the state of each network entity, its connectivity to other network entities and any error or status messages that result from the post-test roll-call function. Messages shall be contained within a scrolled region within the status window, to allow the operator to review multiple messages. The message shall include the entity identifier which reported the information as well as the text of the message.

3.3. CSCI INTERNAL INTERFACE

Figure 2 illustrated both the internal and external interfaces of the CDRSS. Section 3.1 discussed the external interfaces. This section will discuss the requirements for the internal interfaces.

3.3.1. HLA NETWORK INTERFACE TO TEST MONITOR INTERFACE

This interface shall be responsible for reading and writing the HLA attribute and interaction results between the entity and the real-time network. All subscribed data generated externally from an entity shall pass over this interface from the HLA network interface capability to the Test Monitor capability. All data generated by the entity shall pass over this interface from the Test Monitor capability to the HLA network interface capability. The data that passes over this interface in both directions is defined in the data encapsulations for the entity type described in Section 3.4 and the Appendices.

3.3.2. TEST MONITOR TO TEST CONTROL INTERFACE

This interface shall be responsible for transmitting the processed commands from the TCC to the Test Monitor Capability and transmitting network requests and status information from the Test Monitor Capability to the TCC. Commands from the TCC either generated by an operator or as responses to network requests, shall be transmitted to the Test Monitor Capability. These commands include display configurations, data reduction process set-up, and status indications that must be sent over the real-time network to other entities. If these requests force a state change of the entity such as a transition from initialization to test execution, they are sent over this interface to the TCC. If the request or command does not require an operator response, the TCC shall be the only function required to process the command. If an operator response is required, the command or request shall also be transmitted from the Test Monitor Capability to the Data Display Capability over the interface described in Section 3.3.3.

3.3.3. TEST MONITOR TO DATA DISPLAY INTERFACE

This interface shall be responsible for passing information that must be displayed from the Test Monitor Capability to the Data Display Capability. This interface is unidirectional. The Data Display Capability shall not transfer any information back to the Test Monitor Capability. The types of data transmitted over this interface shall be (1) data display control commands, (2) status/network request information, and (3) entity processed data. Commands controlling the types of display, size of displays, and their location on a video monitor are the type of display control commands transmitted over this interface. Requests for entity state changes, network status/health information, entity status/health information, and general test control commands are the types of status/network request information transmitted over this interface. All test data to be displayed by the entity during pre-test, test execution, and post test, are transmitted over this interface as well. During test execution, the data transmissions shall support real-time data rates. Pre-test and post test data transmission rates are not required to be real-time, however, sufficient bandwidth must be made available to prevent data loss.

3.3.4. TEST MONITOR TO DATA REDUCTION INTERFACE

This interface shall be responsible for providing the necessary raw data to the Data Reduction Capability for the TDF configuration and retrieving the reduced data results for both the NE and TDF configurations that perform data reduction. Commands for initializing the data reduction processes shall also be transmitted over this interface to the Data Reduction Capability. The Data Reduction Capability located at the TDF receives its raw data from each network entity connected to the real-time network. This data, after collection by the HLA Network Interface Capability and transmission to the Test Monitor Capability, is passed to the Data Reduction Capability over this interface. The data reduction process for all entities including the TDF shall be initialized by commands received over this interface. In addition, some entities may support multiple data reduction processes. In these cases, this interface shall support commands from the Test Monitor Capability to the Data Reduction Capability indicating which data reduction processes shall be performed. All products from the data reduction processes shall be transmitted over this interface to the Test Monitor Capability. These products will be entity dependent.

3.4. CSCI DATA ELEMENT REQUIREMENTS (SSS 3.2.3.2, 3.7.9.3)

Data element requirements for each entity type have been defined in the System/Subsystem Specification for the real-time network. These data elements were presented in terms of encapsulations for each entity type. An encapsulation represents a single type of entity and the necessary data to interact with this object. Appendices A, B and C to this specification contain the data element definitions from the

System/Subsystem Specification, which forms the requirements for this Software Requirement Specification. Some additional comments are included below to supplement the information provided in the Appendices.

The requirements for interfacing which must occur so that the HLA Interface may connect to the existing entity or entity Facility Control are to be determined locally at each installation. The HLA interface shall be responsible for capturing the current values for each complex attribute data element specified in Appendix C for the given entity class to be interfaced. If the HLA Interface is being developed for a facility serviced by a Facility Control, this HLA Interface shall provide this function for all entities managed by the Facility Control.

The collection of data elements defined in the Appendices for each entity class may be considered a set of structured attributes. In addition to the data type and word length of each data element, the Appendices include a specification for a minimum update interval for each of these structured attributes. The HLA Interface shall publish the entire structured attribute for its entity class at an interval specified in Appendix B.

One allowable exception to this requirement is the *TTRadarPDUStruct* and the *AIRadarPDUStruct* data types. In this case, the missile guidance command data must be updated at 100 Hz. The remaining data (which is the bulk of the attribute) is only needed at 40 Hz. In this case, it is permissible to transmit the command data separate from the remaining radar data on the basis of required update rate.

3.5. ADAPTATION REQUIREMENTS

3.5.1. INSTALLATION DEPENDENT DATA

The adaptation requirements for the installation dependent data for the CDRSS include (1) the HLA Object Model Template (OMT), (2) the Federation Required Execution Details (FRED) (3) entity location and, (4) entity specific installation dependent data. The HLA OMT is necessary to allow the CDRSS to interact with the HLA real-time network. This OMT defines the initial attribute class structures used by the HLA federation. The OMT is an output product of the Federation Development Process (FEDEP), and is unique to each federation. The HLA OMT is used by the HLA Network Interface Capability. The location of the entity in earth-centered coordinates is required by the data reduction capability to perform all necessary coordinate transformations from local coordinate systems to the network earth-centered coordinate system. Specific entity or facility requirements may dictate additional adaptation requirements for installation dependent data. These adaptation requirements for the CDRSS shall be derived from the appropriate facility or entity specifications. The capabilities that may be impacted by these entity specific adaptation requirements include the Data Reduction, Test Control, and Data Display Capabilities.

3.5.2. OPERATIONAL PARAMETERS

There are no operational parameter adaptation requirements for this CSCI. Specific entity or facility requirements may dictate adaptation requirements for operational parameters, however, these will be unique to each facility or entity. Adaptation requirements for the CDRSS shall be derived from the appropriate facility or entity specifications. The capabilities that may be impacted by these entity-specific adaptation requirements include the Data Reduction, Test Control, and Data Display Capabilities.

3.6. SIZING AND TIMING REQUIREMENTS (SSS 3.3.11)

The CDRSS shall be developed such that less than 50% of the available computer resources are utilized. This requirement for 50% utilization applies to processor speed, memory (both RAM and ROM) utilization, and data storage capacity. If the capabilities are developed in a distributed architecture, this requirement shall extend to the expandability of the architecture to support a 100% growth capacity in processor speed, memory utilization and data storage capacity.

3.7. SAFETY REQUIREMENTS

There are no specific safety requirements associated with this CDRSS. Depending on unique entity applications, safety considerations for limiting control functions may be required. These requirements shall be derived from the appropriate entity or facility specifications.

3.8. SECURITY (SSS 3.3.9)

All transmissions of data from a facility or entity shall be encrypted for information security. The real-time network shall support transmission of data at the maximum security level required by any entity participating in the federation. The network shall operate at a single security level, and shall not support multiple levels of security access or security compartmentalization of test assets connected to the network.

3.9. DESIGN CONSTRAINTS

There are no specific design constraints associated with this CDRSS. Depending on unique entity applications, design constraints may exist in the interaction between the CDRSS and the entity or facility. These constraints shall be derived from the appropriate entity or facility specifications.

3.10. SOFTWARE QUALITY FACTORS

The following sections shall present the software quality factor requirements for the CDRSS and the methods of compliance for each.

3.10.1. SOFTWARE RELIABILITY (SSS 3.3.1, 3.2.6.1)

The CDRSS shall not degrade the reliability performance of the attached entities/facilities. Software failures are typically associated with coding errors or ‘bugs’. To achieve the software reliability requirement, these coding errors shall be minimized through the use of a quality software development process. This development process shall at a minimum provide for (1) requirements management, (2) software project planning, tracking, and oversight, (3) software QA, (4) software configuration management, (5) software peer reviews, and (6) integrated software testing. The CDRSS shall whenever possible be developed in a language that provides adequate commercial software development tools. These tools shall be used when available and shall consist of compilers, run-time code monitors/debuggers, and syntax checkers.

3.10.2. SOFTWARE MAINTAINABILITY (SSS 3.3.1, 3.2.6.2)

The CDRSS shall not degrade the maintainability performance of the attached entities/facilities. This requirement shall be met by developing the code using the same quality software development process described for software reliability. In addition, the CDRSS shall be developed under a modular architecture. Adequate documentation and maintenance procedures shall be provided to permit a user to isolate a problem to a particular module in less than 1 hour. Each module shall be limited in size to support the maintainability requirement. Software development tools including a run-time code monitor/debugger shall be available for quickly isolating problems within a module. Whenever possible, these development tools shall be commercially available.

3.10.3. SOFTWARE AVAILABILITY (SSS 3.3.1, 3.2.6.3)

The CDRSS shall not degrade the availability performance of the attached entities/facilities. The requirement for availability shall be met through the same quality software development process as described for software reliability and maintainability.

3.11. HUMAN PERFORMANCE/HUMAN ENGINEERING REQUIREMENTS (SSS 3.6.2)

The design of all CDRSS displays will be in accordance with relevant human engineering standards and accepted practices of human factors engineering. The following documents contain the relevant human engineering standards and describe the human engineering activities required to support the design of the CDRSS.

- MIL-STD-1472D
- MIL-H-46855B

The human engineering program will ensure that all human-machine interfaces are designed to facilitate successful performance of human operator tasks and activities. Within this context the term *human operator* refers to:

1. test control operators
2. test observers (e.g., customer representatives)
3. maintainers of CDRSS

The human engineering program will also ensure appropriate interface design for maintainers. The human engineering program will maximize the efficiency, effectiveness, and safety of all human-in-the-loop aspects of the system. A discussion of human engineering issues relevant to all categories of operators is offered in the following sections.

3.11.1. Human Engineering Issues Relevant to Test Control Operators

Test control operators are those individuals who develop test capabilities, plan test conditions and procedures, initialize and calibrate test equipment, conduct test operations, process test data, and support evaluation of test results. The human engineering program will ensure that all aspects of CDRSS operator-machine interaction are designed to facilitate successful and efficient performance of test operator tasks. Specifically, the software-driven controls and displays will be designed to support all required test control operator functions, which include (but are not limited to) the following:

- Maintain signal library
- Calibrate system components
- Specify test procedures
- Initialize test conditions
- Monitor and control test execution
- Communicate with test observers and other test operators
- Monitor and control test data collection
- Monitor and document exceptions and anomalies
- Interpret test data
- Archive test data
- Analyze test data
- Support evaluation of test data
- Prepare test reports

Test control operators will be provided a common GUI to conduct pre- and post-test asset calibration, perform initialization of the test setup, conduct real-time monitoring of test progress, capture data, and report test results. Through a consistent GUI, information relevant to test construction, test operations, and test anomalies, in addition to asset and performance information, will be provided. Test control operators will also have the ability to display perspectives related to asset configurations, scenario interactions, performance statistics, and test scenarios. Test control operators will be notified when pre-selected test parameter values are not within a pre-selected range.

3.11.2. Human Engineering Issues Relevant to Test Observers

Test observers are those individuals who have a legitimate role in planning, conducting, and evaluating tests but do not directly operate the test equipment. The human engineering program will ensure that all aspects of test customer-machine interaction are designed to facilitate successful completion of all test observer activities.

Test observers will be provided with the ability to observe actual or recorded tests. Interactions between the test observer and test control operators will be facilitated through an integrated, multiple windowing environment (the common GUI). The common GUI will employ technologies that support visualization and ease-of-use through elements such as visual definition tools, interactive icons, menu-based program selection tools, and other GUI-based tools.

3.11.3. HUMAN ENGINEERING ISSUES RELEVANT TO MAINTAINERS

The human engineering program will ensure that all aspects of equipment-maintainer interaction are designed to facilitate successful and efficient performance of maintainer tasks. Physical components will be arranged so that maintainers can access plugs, ports, switches, and connectors safely and efficiently. Physical layout of components will also reflect consideration of requirements for maintainers to remove/replace components and parts in the context of routine preventative maintenance as well as troubleshooting reported problems. Software interfaces for display of test data (e.g., BIT results) will be designed to support the maintainer's information requirements in routine preventative maintenance and in troubleshooting reported problems. Where possible, interaction between maintainer and equipment will be facilitated through the common GUI. Meaningful displays of maintenance and diagnostics data will be presented to the maintainer.

3.11.4. ADDITIONAL HUMAN ENGINEERING ISSUES

Display systems designed for operators and test observers will facilitate the observation of both a global view, in addition to specific views, of the test currently running or of previous tests. The human engineering program will ensure that the design of display systems support the ready interpretation of global and specific views. Software-driven user interfaces will be provided, where prompts, menus, and other aids will facilitate the operation and maintenance of system equipment. User interfaces will be intuitive, standard, uncomplicated, responsive, and flexible. In this manner, the complexity of interface operation will be reduced; the amount of training required for learning interfaces will be reduced; and a means of tailoring interfaces to meet changing mission requirements will be established. Consistency in GUIs and all other human-machine interfaces (with respect to comparable operator actions and commands) will be established. Data entry will be accomplished via a combination of nested menus, menu bars, icons, prompts, or direct input as consistent with the action(s) necessary for a task and the integrated overall design approach for user interfaces. Specific error messages corresponding to unacceptable and inappropriate operator inputs will be provided. Warning messages will be provided prior to execution of any operator commands that could adversely affect the system or system performance.

The control input shall be designed to minimize steps required for executing any command required during the real-time execution of the test. A graduated interface design shall be implemented. The graduation in capability shall permit simplified operation for novice users, at the expense of efficiency. For more advanced users, the interface shall provide less obvious, but more efficient methods for control and display.

Training curriculum, training devices, and training materials must support the ready learning of all required operator tasks and responsibilities. The training program must ensure development of the following operator skills:

1. how to use the systems to develop, conduct, and analyze a test scenario
2. how to operate test assets
3. how to conduct tests
4. how to operate and maintain the configuration
5. how to maintain system software

6. how to provide follow-on training using Government instructors.

3.12. REQUIREMENTS TRACEABILITY

The following table maps the System/Subsystem paragraphs applicable to the CDRSS to the engineering requirement paragraphs presented in this Software Requirements Specification. The mapping of the engineering requirements to the System/Subsystem paragraph has been provided within each requirement header in this document.

Table 1. Requirements Traceability Table

Requirement Description	SSS Paragraph	SRS Paragraph
Test Director Class Functional Description	3.1.1.9	3.2.1, 3.2.1.1, 3.2.1.2, 3.2.1.2.1, 3.2.1.2.2, 3.2.1.3, 3.2.2, 3.2.2.1, 3.2.2.2
Performance Characteristics	3.2.1	3.2.2.1, 3.2.2.2, 3.2.2.3
System Operation	3.2.1.1	3.2.1.1, 3.2.2.1, 3.2.3.1, 3.2.5.1, 3.2.1.2, 3.2.2.2, 3.2.3.2, 3.2.5.2, 3.2.1.3, 3.2.2.3, 3.2.3.3, 3.2.5.3
Performance Characteristics - State 1:Initialization		3.2.1.1, 3.2.2.1, 3.2.3.1, 3.2.5.1
Performance Characteristics - State 2:Execution	3.2.1.3	3.2.1.2, 3.2.2.2, 3.2.3.2, 3.2.5.2
Performance Characteristics - State 3:Post test	3.2.1.4	3.2.1.3, 3.2.2.3, 3.2.3.3, 3.2.5.3
HLA Network Interface	3.2.3.2	3.2.4, 3.4
Data Reduction	3.2.3.4	3.2.3, 3.2.3.1, 3.2.3.2, 3.2.3.2.1, 3.2.3.3, 3.2.5.1, 3.2.5.2, 3.2.5.3
System Reliability	3.2.6.1	3.10.1
System Maintainability	3.2.6.2	3.10.2
System Availability	3.2.6.3	3.10.3
Design and Construction Materials	3.3.1	3.10.1, 3.10.2, 3.10.3
Human Engineering	3.3.7	3.11
System Security	3.3.9	3.8
Sizing and timing requirements	3.3.11	3.6
Personnel and Training	3.6.2	3.11
Test Director Entity - Data Collection Requirements	3.7.9.2.1	3.2.1.2
Test Director Entity - Real-time data collection, processing, analysis, and display	3.7.9.2.3	3.2.5.2
Test Director Entity - Test visualization	3.7.9.2.3.1	3.2.5.1, 3.2.5.2, 3.2.5.3
Test Director Entity - SUT assessment	3.7.9.2.3.2	3.2.3.2, 3.2.3.2.2, 3.2.5.2
User Interface Requirements	3.7.9.2.4	3.2.5
Publication Requirements	3.7.9.3	3.4
Data Bandwidth Requirements	3.7.9.4	3.4
Preparation for Delivery - Software	5.2	5

4. QUALIFICATION REQUIREMENTS

4.1. QUALIFICATION METHODS

The Following table defines the qualification methods (demonstration (D), analysis (A), inspection (I)) for each SRS requirement of the CDRSS.

Table 2. CDRSS Qualification Methods.

Requirement Description	SRS Paragraph	Qualification Method
Test Monitoring	3.2.1	D, I, A
Test Control	3.2.2	D, I
Data Reduction	3.2.3	D, I, A
HLA Network Interface	3.2.4	D, I
Operator Interface	3.2.5	D
Data Element Requirements	3.4	D, I
Adaptation Requirements	3.5	D
Sizing and Timing Requirements	3.6	A
Security Requirements	3.8	D
Software Reliability	3.10.1	D, I
Software Maintainability	3.10.2	D, I
Software Availability	3.10.3	A
Human Performance/ Human Engineering	3.11	D
Packaging	5	I

4.2. SPECIAL QUALIFICATION REQUIREMENTS

There are no special qualification test requirements for this CSCI. Acceptance Test Procedures for verifying requirements compliance of the CDRSS shall not require the development of any specialized tools. The CDRSS shall provide all necessary information to evaluate specification compliance.

5. PREPARATION FOR DELIVERY (SSS 5.2)

All magnetic media containing source and/or object code for the CDRSS and program code listings shall be labeled as follows:

Program/ System Name
Program Identification Number
Version Identifier
Version Date

The Version scheme described above shall be in accordance with the TSLA system program configuration and data management plan as applicable to software delivery.

6. NOTES

6.1. LIST OF ACRONYMS

ADS	Advanced Distributed Simulation
API	Application Program Interface
BIT	Built In Test
CDRSS	Control, Display, Reduction Software Segment
CSCI	Computer Software Configuration Item
ECM	Electronic Countermeasure
EW	Electronic Warfare
FEDEP	Federation Development Process
FRED	Federation Required Execution Details
GUI	Graphical User Interface
HLA	High Level Architecture
MOE	Measure of Effectiveness
MOP	Measure of Performance
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
NE	Node Executive
OMT	Object Model Template
QA	Quality Assurance
RAM	Random Access Memory
RF	Radio Frequency
ROM	Read Only Memory
RTI	Run-Time Infrastructure
SPP	System Performance Parameter
SUT	System Under Test
T&E	Test and Evaluation
TCC	Test Control Capability
TDF	Test Director Facility
TPP	Technical Performance Parameter
TSLA	Threat Simulation Linking Activities
TSPI	Time-Space-Position Information

APPENDIX A. TSLA OBJECT STRUCTURE CLASS TABLE

Object Class Structure Table		
Entity	Radar Base	EW Radar
		HF Radar
		TA Radar
		TT Radar
		AI Radar
		MW Radar
	Threat Base	Command Guided Missile
		Active Missile
		Artillery
		SemiActive Missile
	Warning Base	Radar Warning Receiver
	ECM Base	Jammer
		Towed Jammer
		Chaff Dispenser
	C3	
	Stand Alone Monitor	
	Comm ECM	
	Platform	
Facility Controller		
Test Director		

APPENDIX B. TSLA FOM ATTRIBUTE PARAMETER TABLE

Attribute/Parameter Table										
Object/Interaction	**Attribute/Parameter**	**Datatype**	**Card**	**Units**	**Res**	**Accuracy**	**Accuracy Condition**	**Update Type**	**Update Condition**	**Bytes**
Entity.Radar_Base.EW_Radar	EWRadarPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	53
Entity.Radar_Base.HF_Radar	HFRadarPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	53
Entity.Radar_Base.TA_Radar	TARadarPDUStruct	Complex Data Type						conditional	0.05 sec update interval passed	53
Entity.Radar_Base.TT_Radar	TTRadarPDUStruct	Complex Data Type						conditional	0.025 sec update interval passed 0.01 sec for Guidance	151
Entity.Radar_Base.AI_Radar	AIRadarPDUStruct	Complex Data Type						conditional	0.025 sec update interval passed 0.01 sec for Guidance	135
Entity.Radar_Base.MW_Radar	MWRadarPDUStruct	Complex Data Type						conditional	0.025 sec update interval passed	37
Entity.ThreatBase.CommandGuidedMissile	CommandGuidedMissilePDUStruct	Complex Data Type						conditional	0.025 sec update interval passed	71
Entity.ThreatBase.ActiveMissile	ActiveMissilePDUStruct	Complex Data Type						conditional	0.025 sec update interval passed	71
Entity.ThreatBase.Artillery	ArtilleryPDUStruct	Complex Data Type						conditional	0.025 sec update interval passed	71
Entity.ThreatBase.SemiActiveMissile	SemiActiveMissilePDUStruct	Complex Data Type						conditional	0.025 sec update interval passed	71
TestDirector	TestDirectorPDUStruct	Complex Data Type						conditional	1.0 sec update interval passed	46
FacilityController	FacilityControllerPDUStruct	Complex Data Type						conditional	1.0 sec update interval passed	46
Entity.WarningBase.RadarWarningReceiver	RadarWarningReceiverPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	17
Entity.ECMBase.ChaffDispenser	ChaffDispenserPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	90
Entity.ECMBase.Jammer	JammerPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	93
Entity.ECMBase.TowedJammer	TowedJammerPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	105
Entity.C3	C3PDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	86
Entity.StandAloneEnvironmentMonitor	StandAloneEnvironmentMonitorPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	15
Entity.CommunicationECM	CommunicationECMPDUStruct	Complex Data Type						conditional	0.1 sec update interval passed	89
Entity.Platform	PlatformPDUStruct	Complex Data Type						conditional	0.025 sec update interval passed	104

APPENDIX C. TSLA COMPLEX DATATYPE TABLE

See TSLA Network Requirements Specification 28 February 1997, Appendix A for a description of the information in each of these complex data types.

Complex Datatype Table

Complex Datatype	Field	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condtion
EWRadarPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	local_target_tag	int					
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	scan_period	long		ms	1		
	azimuth_valid	bool					
	elevation_valid	bool					
	range_valid	bool					
	doppler_valid	bool					
	injection_verified	bool					
	emission_verified	bool					
	begin_pulse	bool					
	checksum	unsigned long					
HFRadarPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	local_target_tag	int					
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	op_command	Complex Type					
	nod_period	long		ms	1		
	azimuth_valid	bool					
	elevation_valid	bool					
	range_valid	bool					
	doppler_valid	bool					
	injection_verified	bool					
	emission_verified	bool					
	begin_pulse	bool					
	checksum	unsigned long					
TARadarPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	local_target_tag	int					
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	boresight_azimuth	int		deg	0.1		
	boresight_elevation	int		deg	0.1		
	azimuth_valid	bool					
	elevation_valid	bool					
	range_valid	bool					
	doppler_valid	bool					
	injection_verified	bool					
	emission_verified	bool					
	checksum	unsigned long					
TTRadarPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	local_target_tag	int					
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	boresight_azimuth	int		deg	0.1		
	boresight_elevation	int		deg	0.1		
	track_error_azimuth	int		deg	0.1		
	track_error_elevation	int		deg	0.1		
	track_error_range	long		m	0.1		
	jammer_to_signal_ratio	int		db	0.1		
	signal_to_clutter_ratio	int		db	0.1		
	op_command	Complex Type					
	op_response	Complex Type					
	engagement_result	enum	EngagementResultEnum				
	miss_distance_x	long		m	0.1		
	miss_distance_y	long		m	0.1		
	miss_distance_z	long		m	0.1		
	azimuth_valid	bool					
	elevation_valid	bool					
	range_valid	bool					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	doppler_valid	bool					
	injection_verified	bool					
	emission_verified	bool					
	track_status	bool					
	checksum	unsigned long					
AIRadarPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	local_target_tag	int					
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	platform_serial_number	int					
	boresight_azimuth	int		deg	0.1		
	boresight_elevation	int		deg	0.1		
	track_error_azimuth	int		deg	0.1		
	track_error_elevation	int		deg	0.1		
	track_error_range	long		m	0.1		
	jammer_to_signal_ratio	int		db	0.1		
	signal_to_clutter_ratio	int		db	0.1		
	op_command	Complex Type					
	op_response	Complex Type					
	engagement_result	enum	EngagementResultEnum				

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	miss_distance_x	long		m	0.1		
	miss_distance_y	long		m	0.1		
	miss_distance_z	long		m	0.1		
	azimuth_valid	bool					
	elevation_valid	bool					
	range_valid	bool					
	doppler_valid	bool					
	injection_verified	bool					
	emission_verified	bool					
	track_status	bool					
	checksum	unsigned long					
MWRadarPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	local_target_tag	int					
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	platform_serial_number	int					
	boresight_azimuth	int		deg	0.1		
	boresight_elevation	int		deg	0.1		
	azimuth_valid	bool					
	elevation_valid	bool					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	range_valid	bool					
	doppler_valid	bool					
	injection_verified	bool					
	emission_verified	bool					
	checksum	unsigned long					
CommandGuidedMissilePDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	radar_serial_number	int					
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	op_response	Complex Type					
	armed	bool					
	launch_status	bool					
	target_lock	bool					
	checksum	unsigned long					
ActiveMissilePDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	radar_serial_number	int					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	op_response	Complex Type					
	armed	bool					
	launch_status	bool					
	target_lock	bool					
	checksum	unsigned long					
ArtilleryPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	radar_serial_number	int					
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	op_response	Complex Type					
	armed	bool					
	launch_status	bool					
	target_lock	bool					
	checksum	unsigned long					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
SemiActivePDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	radar_serial_number	int					
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	op_response	Complex Type					
	armed	bool					
	launch_status	bool					
	target_lock	bool					
	checksum	unsigned long					
TestDirectorPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	director_command	Complex Type					
	director_response	Complex Type					
	checksum	unsigned long					
FacilityControllerPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	director_command	Complex Type					
	director_response	Complex Type					
	checksum	unsigned long					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
RadarWarningPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	platform_serial_number	int					
	warning	bool					
	emission_verified	bool					
	checksum	unsigned long					
ChaffDispenserPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	platform_serial_number	int					
	op_command	Complex Type					
	op_response	Complex Type					
	checksum	unsigned long					
JammerPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	power_percent	int		%	0.1		
	platform_serial_number	int					
	op_command	Complex Type					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	op_response	Complex Type					
	injection_verified	bool					
	emission_verified	bool					
	checksum	unsigned long					
TowedJammerPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	mode_serial_number	int					
	power_percent	int		%	0.1		
	platform_serial_number	int					
	position_offset_x	long		m	0.1		
	position_offset_y	long		m	0.1		
	position_offset_z	long		m	0.1		
	op_command	Complex Type					
	op_response	Complex Type					
	injection_verified	bool					
	emission_verified	bool					
	checksum	unsigned long					
C3PDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	op_command	Complex Type					
	op_response	Complex Type					
	checksum	unsigned long					

Complex Datatype Table

Complex Datatype	Field	Datatype	Cardinality	Units	Resolution	Accuracy	Accuracy Condtion
StandAloneEnvironmentMonitorPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	verified	bool					
	checksum	unsigned long					
CommunicationECMPDUStruct	SerialNumber	int					
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	azimuth	int		deg	0.1		
	elevation	int		deg	0.1		
	range	long		m	0.1		
	doppler	long		Hz	1		
	op_command	Complex Type					
	op_response	Complex Type					
	engagement_result_enum	enum					
	azimuth_valid	bool					
	elevation_valid	bool					
	range_valid	bool					
	doppler_valid	bool					
	armed	bool					
	launch_status	bool					
	target_lock	bool					
	checksum	unsigned long					
PlatformPDUStruct	SerialNumber	int					

Complex Datatype Table							
Complex Datatype	**Field**	**Datatype**	**Cardinality**	**Units**	**Resolution**	**Accuracy**	**Accuracy Condtion**
	time	unsigned long		ms	1		
	entity_status	enum	EntityStatusEnum				
	communication_status	enum	CommunicationStatusEnum				
	position_x	long		m	0.1		
	position_y	long		m	0.1		
	position_z	long		m	0.1		
	orientation_phi	int		deg	0.1		
	orientation_theta	int		deg	0.1		
	orientation_psi	int		deg	0.1		
	op_command	Complex Type					
	op_response	Complex Type					
	checksum	unsigned long					
OpCommandStruct	op_code	enum	OpCodeEnum				
	data	char[34]					
	checksum	unsigned long					
OpResponseStruct	op_code	enum	OpCodeEnum				
	data	char[34]					
	checksum	unsigned long					
DirCommandStruct	op_code	enum	OpCodeEnum				
	data	char[16]					
	checksum	unsigned long					
DirResponseStruct	op_code	enum	OpCodeEnum				
	data	char[16]					
	checksum	unsigned long					

APPENDIX D. TSLA ENUMERATED DATA TYPES

Enumerated Datatype Table		
Identifier	**Enumerator**	**Representation**
CategoryEnum	CATEGORY_UNKNOWN	0
CommunicationStatusEnum	COMM_OKAY COMM_LATE COMM_FAULT	0 1 2
CountryEnum	COUNTRY_UNKNOWN	0
DirOpCodeEnum	DIR_NOP	0
DomainEnum	DOMAIN_UNKNOWN	0
ECMTechniqueEnum	ECM_UNKNOWN ECM_NOISE ECM_FALSE_TARGET	0 1 2
EngagementResultEnum	ENGAGE_UNKNOWN ENGAGE_PENDING ENGAGE_ENGAGING ENGAGE_KILL ENGAGE_NO_KILL	0 1 2 3 4
EngagementStatusEnum	ENGAGE_NOT_ENGAGED ENGAGE_ENGAGE	0 1
EntityKindEnum	KIND_UNKNOWN	0
EntityStatusEnum	ENTITY_ON ENTITY_OFF ENTITY_STANDBY ENTITY_FAIL	0 1 2 3
ExtraEnum	EXTRA_UNKNOWN	0
OpCodeEnum	OP_NOP OP_ARM OP_ASSIGN OP_DETONATE OP_GUIDE OP_HANDOFF OP_INTERCEPT OP_LAUNCH OP_SWITCH OP_SENDLOCKSTATUS	0 1 2 3 4 5 6 7 8 9

Enumerated Datatype Table		
Identifier	**Enumerator**	**Representation**
	OP_SENDTARGETZONE	10
	OP_CHAFF_DISPENSE	11
RadarTypeEnum	RADAR_UNKNOWN	0
	RADAR_EW	1
	RADAR_HF	2
	RADAR_TAR	3
	RADAR_TTR	4
	RADAR_AIR	5
	RADAR_MWR	6
RadarOperationEnum	RADAROP_UNKNOWN	0
	RADAROP_SCAN	1
	RADAROP_TRACK	2
	RADAROP_HEIGHTFIND	3
SubCategoryEnum	SUBCATEGORY_UNKNOWN	0
SpecificEnum	SPECIFIC_UNKNOWN	0

APPENDIX E. LATENCY COMPENSATION METHOD

E.1. PERFORMANCE REQUIREMENTS

The distributed nature of the test network, and the processing time of the RTI will insert some amount of latency in the delivery of data to the end user. For the purposes of this specification, latency is defined as the difference in time between when data are measured by the data generator and when the data reaches the data consumer. The data generator shall time tag the data to correspond to the time that the measurement was made. The data consumer shall record the time the data was provided. The time difference is the latency.

The approach to latency compensation is to estimate the *state* of the dynamic system (e.g. the platform position and velocity in a 6-state model, or its position, velocity and acceleration in a 9-state model) and then use these estimates to predict its future position. If the state estimates are accurate when transmitted, then predictions based on a kinematic extrapolation will remain accurate until the highest derivative of the target state that is used in the predictor changes from the value in the transmitted estimate.

E.1.1. LATENCY RANGE

The expected range of values for latency in the test network is difficult to determine reliably. Also, it is recognized that latency compensation can only be performed for those data sources where the highest derivatives in the underlying system model remain constant during the interval of compensation. (The state variables include things like *position* which need not remain constant (of course). What must remain constant is either the system model, or perhaps the highest derivative modeled). For flight encounters of interest, it would be unreasonable to expect that the highest order derivative would remain constant for longer than 0.5 seconds. Hence, on this basis, the maximum range of compensation for latency shall be 0.5 seconds.

E.1.2. PREDICTION ACCURACY

The accuracy of prediction following latency compensation shall be less than 0.5 meters for range and less than 0.1 milliradian for azimuth and elevation.

E.2. DATA UPDATE INTERVAL

The required update interval is dictated by a) the actual data transmitted from generator to consumer, and b) the minimum expected interval of time over which the highest derivatives of the state variables can be expected to remain constant. For the platforms of interest to EW testing (i.e., tactical aircraft and missiles) the maximum interval over which these state variables can be expected to remain constant is 25ms. This shall be the maximum update interval for state estimates.

E.3. COMPUTATION OF STATE ESTIMATES

The TSLA network shall include, at each data generator, a filter capable of producing estimates of the current (3-space) position of the platform, and the first time derivatives of these parameters. The updates of these estimates shall be provided at a 40Hz rate. The TSLA network shall require that a predictor algorithm be implemented at each data consumer (where latency compensation is required) which will operate with the state estimates to produce predictions with accuracy as specified in Section E.1. If there is a requirement for raw (unfiltered data) these data values shall be transmitted within the same data structure as that for the filtered state estimates.

The rationale for this specification is provided in the paragraphs below.

There are two parts to the latency compensation process. The first part is a filtering process which produces accurate estimates of the system state variables. The second part is an extrapolation process which uses the state estimates to form a predicted value of position based on the state estimates. The prediction computation is relative to the time of the state estimates. The time for the prediction is the current time.

There are three possible implementations of the filter-predictor process. In the first two methods, the filter is located at the data generator, while predictor is resident at the data consumer. There are two cases here because, with the filter, it is possible to provide updates of the state estimates at a rate different from the sample interval provided by the measurement sensor. The data generator can monitor test the results of predictions which would be made by the current state estimates, and only provide updates to the state estimates when these predictions exceeded some error tolerance. Or, in the second approach, the data generator could simply provide a state update each time there was a new measurement. The first approach provides the opportunity to reduce the amount of data traffic. But, it requires the data generator to test for the condition when the prediction error becomes too large. Also, this approach has the limitation that the state is updated only after it is determined that the current state estimate will produce excessive prediction error. Hence, there is an inherent latency built in to this approach.

The third approach places both the filter and the predictor at the data consumer node. This approach is appealing whenever there is a need for the raw sensor data to be transmitted over the network (For example if it is required for use in test data reduction.) In this situation, both the filter and the predictor are implemented at the data consumer node. Here, the drawback is that each data consumer must now support the computational burden of the filter.